ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Computational Geometry 39 (2008) 163–177
Computational Geometry
Theory and Applications
www.elsevier.com/locate/comgeo

Visibility maps of segments and triangles in 3D

Esther Moet [a,*], Christian Knauer [b], Marc van Kreveld [b]

[a] *Department of Information and Computing Sciences, Universiteit Utrecht, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands*
[b] *Institute of Computer Science, Freie Universität Berlin, Takustraße 9, D-14195 Berlin, Germany*

Received 20 January 2006; received in revised form 3 November 2006; accepted 6 November 2006

Available online 15 December 2006

Communicated by D. Avis

Abstract

Let T be a set of n triangles in three-dimensional space, let s be a line segment, and let t be a triangle, both disjoint from T. We consider the subdivision of T based on (in)visibility from s; this is the *visibility map* of the segment s with respect to T. The visibility map of the triangle t is defined analogously. We look at two different notions of visibility: strong (complete) visibility, and weak (partial) visibility. The trivial $\Omega(n^2)$ lower bound for the combinatorial complexity of the strong visibility map of both s and t is almost tight: we prove an $O(n^2\alpha(n))$ upper bound for both structures, where $\alpha(n)$ is the extremely slowly increasing inverse Ackermann function. Furthermore, we prove that the weak visibility map of s has complexity $\Theta(n^5)$, and the weak visibility map of t has complexity $\Theta(n^7)$. If T is a polyhedral terrain, the complexity of the weak visibility map is $\Omega(n^4)$ and $O(n^5)$, both for a segment and a triangle. We also present efficient algorithms to compute all discussed structures.

Keywords: Visibility; Aspect graph; Polyhedral terrains; Combinatorial complexity

1. Introduction

Computations concerning visibility and the generation of shadows are important to obtain realistic images in computer graphics. Unfortunately, these computations can be very time-consuming. The *visibility map* of a point p with respect to a set T of n triangles in 3D, which we call a *3D scene*, is the set of maximally connected components on T that are visible from p.

Definition 1. Let T be a 3D scene, and let p be a point in $\mathbb{R}^3$. We denote by $\mathrm{VP}(p, T)$ the visibility polyhedron of p with respect to T, i.e., the connected subset of $\mathbb{R}^3$ that contains all points that see p. The subdivision of the triangles of T induced by $\mathrm{VP}(p, T)$ is the *visibility map* of p and is denoted by $\mathrm{VM}(p, T)$.

Through (perspective) projection onto a viewplane near p, the visibility map of the point p directly corresponds to a planar graph with $\Theta(n^2)$ vertices, edges, and faces [18,20]; see Fig. 1. In this paper, we study the worst-case

* Corresponding author.
E-mail addresses: esther@cs.uu.nl (E. Moet), christian.knauer@inf.fu-berlin.de (C. Knauer), marc@cs.uu.nl (M. van Kreveld).

0925-7721/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comgeo.2006.11.001

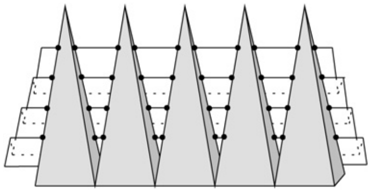


Fig. 1. A visibility map of a point with $\Omega(n^2)$ vertices.

combinatorial complexity of the visibility map for a set that consists of *more than a single point*; in particular, we consider the cases where the view element is a line segment or a triangle. Note that the above graph analogy is no longer valid in these cases. In the definitions below, s is a low-dimensional simplex, for example, a line segment or a triangle in $\mathbb{R}^3$.

Definition 2. The *strong visibility polyhedron* of s, denoted by $\mathrm{SVP}(s, T)$, is the set of points in $\mathbb{R}^3$ that see every point in s. The subdivision of the triangles of T induced by $\mathrm{SVP}(s, T)$ is the *strong visibility map* of s; we denote this structure by $\mathrm{SVM}(s, T)$.

Definition 3. The *weak visibility polyhedron* of s, denoted by $\mathrm{WVP}(s, T)$, is the set of points in $\mathbb{R}^3$ that see at least one point of s. The subdivision of the triangles of T induced by $\mathrm{WVP}(s, T)$ is the *weak visibility map* of s; we denote this structure by $\mathrm{WVM}(s, T)$.

In general, visibility polyhedra are connected sets, whereas visibility maps consist of multiple connected components. The *strong visibility map* of S identifies the maximally connected components on T in which the points see *every* point of s, whereas the other points of T do not see all of s, i.e., they see s partly or not at all. The *weak visibility map* separates the points that see at least one point in S, and the points that do not see any point in s. An alternative way of looking at the weak visibility map is to consider s to be a light source. Then the weak visibility map of s is equivalent to the set of maximally connected components on T that are illuminated; we will sometimes call the weak visibility map of s the *shadow map* of s.

With the results of Wang and Zhu [26], the weak visibility map of a triangle in a general 3D scene can be computed in $O(n^9)$ time and $O(n^7)$ space; to compute the same structure in a polyhedral terrain, the algorithm of [26] can be used to achieve a running time of $O(n^7)$ time and $O(n^5)$ storage. In this paper, we improve significantly on these previous results and show that our bounds are almost tight by giving lower bound constructions. For a line (segment), a collection of general results can be found in [3]. Both of the above papers use the detection of transparent and opaque topology changes to compute the parts of a 3D scene that are weakly visible from a given view element; these topology changes are generally known as EEE events [12,21].

A summary of the most important related results in computational geometry can be found in [20]. Recently, an $O(n^2 \log n)$ time algorithm was presented to compute the weak visibility map of an edge in a terrain restricted to another edge [6]. One of the main topics related to our study is the *aspect graph* [17]. This graph corresponds directly to the more frequently used partition of $\mathbb{R}^3$ into three-dimensional cells such that the visibility map with respect to the scene is combinatorially the same throughout each cell. The *critical points* where the view of the scene changes combinatorially lie exclusively on *critical surfaces* called reguli; see [12] and Section 2. The above mentioned spatial subdivision is induced by a set of $O(n^3)$ of these critical surfaces, and therefore has complexity $O(n^9)$. However, it has been shown that its complexity is almost a linear factor less for polyhedral terrains [1,7].

Shadows are of major interest in computer graphics. The strong and weak visibility polyhedron of a view element are closely related to the *antiumbra* and *antipenumbra* of a light source, respectively. Many papers consider computing the shadow from a given point light source, either in image space, see [27], or in object space [5,11]. The exact shadow of a segment or area light source is considered too complex to compute exactly. This may be the reason that a majority of the studies until now have been incomplete; some address a particular type of 3D scene, e.g., a sequence of convex areal holes [25], others give algorithms that yield an approximation of the shadow [5,6,8,14].

Table 1 gives a summary of our results; for completeness a previous result is included. All displayed results are worst-case combinatorial complexities, and $\alpha(n)$ is the extremely slowly increasing inverse Ackermann function. We

Table 1
Combinatorial complexities of the visibility map in a 3D scene for various types of view elements under different models of visibility

	Strong	Weak (terrain)	Weak (general)	
Point		$\Theta(n^2)$		[18]
Segment	$\Omega(n^2)$	$\Omega(n^4)$ and $O(n^5)$	$\Theta(n^5)$	Sections 3 and 4
Triangle	$O(n^2\alpha(n))$	$\Omega(n^4)$ and $O(n^5)$	$\Theta(n^7)$	Section 5

also present algorithms to compute the discussed structures within almost matching time bounds, i.e., in most cases the computation time is only an $O(\log n)$ factor worse than the worst-case complexity.

This paper is organized as follows. In Section 2, we give several definitions and assumptions. We address the strong and the weak visibility map of an edge in Section 3 and 4, respectively. In Section 5, we study the complexity of the two types of visibility maps when the view element is a triangle, and we round up in Section 6 with the conclusions and a discussion of the obtained results.

2. Preliminaries

In this paper, we consider sets of n triangles in $\mathbb{R}^3$ with pairwise disjoint interiors. We call such a set T a *3D scene*, or simply *scene*. Without loss of generality, we assume that the vertical direction is the z-direction. A special case of a scene is a *polyhedral terrain*, or simply *terrain*, in which T is a piecewise-linear continuous function defined over the triangles of a triangulation in the xy-plane. Given a scene T and two points $p, q \in \mathbb{R}^3$, we say that *p sees q* if the open line segment pq does not intersect the interior of any triangle in T, i.e., grazing contact of the line-of-sight with the edges of the triangles of T is permitted. In the case of terrains, two points can only be mutually visible if they both lie on or above the terrain. All our results, both the bounds on the worst-case complexities and the algorithms, hold when T contains degeneracies. However, to simplify argumentation, we only consider general position configurations in our proofs. This is no restriction; the worst-case complexities can be achieved by a 3D scene in general position, and in the presented algorithms, we can apply standard symbolic perturbation techniques [9].

To study visibility in $\mathbb{R}^3$, it is important to consider collections of lines that intersect three given lines. Such collections are called *reguli*. More formally, the regulus $\mathcal{R}(l_1, l_2, l_3)$ for three lines l_1, l_2, and l_3, is the set of all lines that intersect l_1, l_2, and l_3, in this (or the inverse) order. A regulus has the following properties [22, §3]:

1. it is one ruling of either a hyperbolic paraboloid, or a hyperboloid of one sheet;
2. it is an algebraic surface of constant degree;
3. the projection of its boundary onto the xy-plane is a set of algebraic curves of constant degree;
4. it can be partitioned into a constant number of xy-monotone parts.

Let $\ell(s)$ denote the supporting line of a line segment s. For three edges e, f, and g of T, all lines that intersect e, f, and g, in this (or the inverse) order, lie in a subset of $\mathcal{R}(\ell(e), \ell(f), \ell(g))$. This set consists of at most three connected components [4], and is denoted by $\mathcal{R}(e, f, g)$. The surface $\mathcal{R}(e, f, g)$ can be computed in $O(1)$ time [4]. The critical points where the view of the scene changes combinatorially lie exclusively on these reguli [12].

Throughout this paper, we adopt an appropriate model of computation, i.e., we assume that computing various elementary operations on algebraic curves and surfaces of constant degree can be performed in constant time; for a discussion of computational real algebraic geometry, see for instance [15].

Let e be an edge of T, and let t be a triangle of T. In this paper, we study the worst-case combinatorial complexities of the following four structures: SVM(e, T), WVM(e, T), SVM(t, T), and WVM(t, T). We assume that e or t is a feature of T, but this is no restriction; if an arbitrary segment or triangle (1- or 2-simplex) $s \subset \mathbb{R}^3$ does not intersect any triangle in T, and lies above T in case T is a polyhedral terrain, then we can transform s into a feature s' of a 3D scene T' in $O(n \log n)$ time in such a way that the visibility map of s in T is essentially the same as the visibility map of s' in T'. If T is a general 3D scene, this is straightforward. In the case of a triangular view element, we simply add this triangle to obtain the scene T'. For a line segment s, we create an additional skinny triangle that has s as an edge. The transformation is a little bit more involved when T is a polyhedral terrain. If the view element is a line segment

s, which lies entirely above T, we build a near-vertical wall below s, and we triangulate the affected triangles of T to obtain the transformed terrain T'. If the view element is a triangle t that lies entirely above T, we consider the two types of visibility map separately. Because every point on T that lies below t is weakly visible from t, we can treat this part of the terrain separately and connect t downwards (again, near-vertical) with T to obtain T'. For strong visibility, we cannot do this, since it is not trivial to compute the strong visibility map of t for the part of T below t. In this case, we create a general 3D scene T', which is no longer a polyhedral terrain, that consists of t plus all the triangles of T. This is allowed, because the combinatorial complexity of the strong visibility map of a triangle is the same in a general 3D scene as it is in a polyhedral terrain (see Section 3).

As mentioned in the introduction, the strong and weak visibility polyhedron are closely related to the *antiumbra* and *antipenumbra* of a light source, respectively, which are well-known structures in computer graphics. More precisely, the antiumbra is the volume from which all points on the light source can be seen, and the antipenumbra is the volume from which some, but not all, of the light source can be seen [25]. In this setting, the strong visibility polyhedron is equal to the antiumbra, and the weak visibility polyhedron is equal to the closure of the union of the antiumbra and antipenumbra.

A trivial lower bound for the complexity of all four types of visibility maps is $\Omega(n^2)$, which is a lower bound for the complexity of the visibility map of a single point [18].

Definition 4. [17] Let T be a 3D scene. The *aspect graph* induced by, and restricted to, T is the subdivision of the triangles of $\mathbb{R}^3$ into cells in which the perspective projection of T is combinatorially the same.

The points in one cell of the aspect graph see exactly the same set of features of T; in particular, throughout each cell the simplex s is either visible or invisible. Thus, the complexity of the aspect graph restricted to T is an upper bound on both $\mathrm{SVM}(s, T)$ and $\mathrm{WVM}(s, T)$. On each of the n triangles of T, the aspect graph is defined by the arrangement of $O(n^3)$ critical surfaces, which has complexity $O(n^6)$, and thus the overall complexity is $O(n^7)$. Hence, the complexity of each of the above defined visibility maps is $\Omega(n^2)$ and $O(n^7)$; these bounds also hold for visibility maps that we do not discuss here, e.g., the shadow map of a polyhedral light source.

3. The strong visibility map of an edge

The 2D counterpart of the strong visibility polyhedron of an edge in a 3D scene is the strong (or alternatively, complete) segment visibility polygon $\mathrm{SVP}(e, P)$ for an edge e in a simple polygon P, where P may have holes. This structure has complexity $\Theta(n)$ [2, p. 837]. In fact, the strong visibility polygon of a segment pq within P is equal to the visibility polygon of the point p *within* the visibility polygon of the point q in P, i.e., $\mathrm{SVP}(pq, P) = \mathrm{VP}(p, \mathrm{VP}(q, P))$.

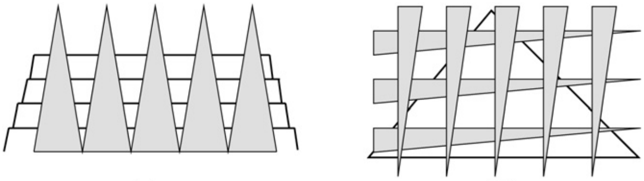
Before we prove a lower and upper bound on the complexity of $\mathrm{SVM}(e)$, we prove that the analog of the above holds in three dimensions as well. By Δ_{sp}, we denote the triangle with base s and apex p.

Lemma 5. *Let T be a 3D scene, and let $e = vw$ be an edge of T. The strong visibility polyhedron* $\mathrm{SVP}(e, T)$ *is equal to* $\mathrm{VP}(v, \mathrm{VP}(w, T))$.

Proof. First, we show that if $p \in \mathrm{SVP}(e, T)$ then $p \in \mathrm{VP}(v, \mathrm{VP}(w, T))$. The triangle Δ_{ep} does not intersect any triangle of T. In particular, the line segment vp does not intersect T, so $p \in \mathrm{VP}(v, T)$. Because the whole of Δ_{ep} is free from intersections with T, v sees all points on the line segment wp as well. This means that the line segment $wp \subset \mathrm{VP}(v, T)$ and thus that w sees p within $\mathrm{VP}(v, T)$.

Second, we show that if $p \in \mathrm{VP}(v, \mathrm{VP}(w, T))$ then $p \in \mathrm{SVP}(e, T)$. We have that p sees both vertices v and w, which means the line segments vp and wp are not intersected by T. Furthermore, both these line segments, and the edge e, lie in $\mathrm{VP}(v, T)$, which is a star-shaped polyhedron. This immediately implies that the triangle Δ_{ep} completely lies in $\mathrm{VP}(v, T)$. This means that Δ_{ep} is free from intersections with T, and thus $p \in \mathrm{SVP}(e, T)$. □

Unfortunately, this analysis does not provide us with a good upper bound on the complexity of $\mathrm{SVP}(e, T)$, since the visibility polyhedron of a point in a 3D scene already has complexity $\Theta(n^2)$ in the worst case, and the visi-



(a) (b)

Fig. 2. (a) A terrain that induces a strong edge visibility map of total complexity $\Omega(n^2)$. (b) A single triangle in a 3D scene (no polyhedral terrain) with a strong edge visibility map of $\Omega(n^2)$ on it. In both figures, the shaded triangles in the front induce edges and vertices on the white triangle(s) in the background.

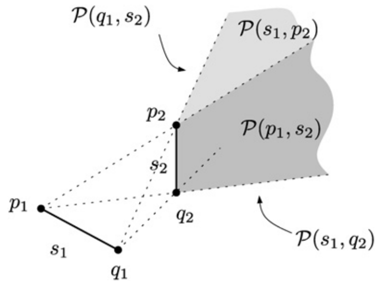


Fig. 3. Illustration to Definition 7.

bility polyhedron of a point in a scene with $O(n^2)$ triangles has complexity $O(n^4)$. Therefore, we take a different approach.

In the remainder of this section, we are given a 3D scene T with n triangles and an edge e of T, and we aim to bound the complexity of the strong visibility map $\mathrm{SVM}(e, T)$ of e on T. A trivial lower bound for $\mathrm{SVM}(e, T)$ is $\Omega(n^2)$; see Section 2 and Fig. 2.

Lemma 6. *The strong visibility polyhedron* $\mathrm{SVP}(e, T)$ *of an edge e in a 3D scene T is connected.*

Proof. Let p and q be two points in $\mathbb{R}^3$ that strongly see e; this implies that the interiors of the triangles Δ_{ep} and Δ_{eq} are free from intersections with the interiors of the triangles in T. Let x be any point on e. The two line segments px and qx are free from intersections with T as well. Furthermore, for every point y on either px or qx, the triangle Δ_{ey} does not intersect T, and thus y is in $\mathrm{SVP}(e, T)$. Therefore, $\mathrm{SVP}(e, T)$ is connected. □

Since the triangles of T already induce $\mathrm{SVP}(e, T)$, the complexity of $\mathrm{SVM}(e, T)$ is at most the complexity of $\mathrm{SVP}(e, T)$. We analyze the complexity of $\mathrm{SVP}(e, T)$ to obtain an almost tight $O(n^2\alpha(n))$ upper bound on the complexity of $\mathrm{SVM}(e, T)$.

Definition 7. Let p be a point, and let s be a line segment, both in $\mathbb{R}^3$. We define $\mathcal{P}(p, s)$ to be the set of points q for which there exists a ray starting at p that first passes through s and then through q. Similarly, $\mathcal{P}(s, p)$ is the set of points q for which there exists a ray starting at a point on s that first passes through p and then through q.

For two line segments $s_1 = p_1q_1$ and $s_2 = p_2q_2$, the (unbounded) region enclosed by $\mathcal{P}(s_1, p_2)$, $\mathcal{P}(p_1, s_2)$, $\mathcal{P}(q_1, s_2)$, and $\mathcal{P}(s_1, q_2)$ contains exactly the points p for which the triangle with base s_1 and apex p intersects the segment s_2; see Fig. 3. For our purposes, if v is a vertex and e an edge of T, where v and e are not incident to each other, we represent the unbounded object $\mathcal{P}(e, v)$ by a large triangle, by clipping it with a vertical plane that lies far enough such that we do not lose any features of $\mathrm{SVP}(e, T)$. Furthermore, we clip $\mathcal{P}(v, e)$ in the same way and represent it by a finite-sized quadrilateral. From now on, we consider a triangle to be a degenerate quadrilateral, in which one side has zero length.

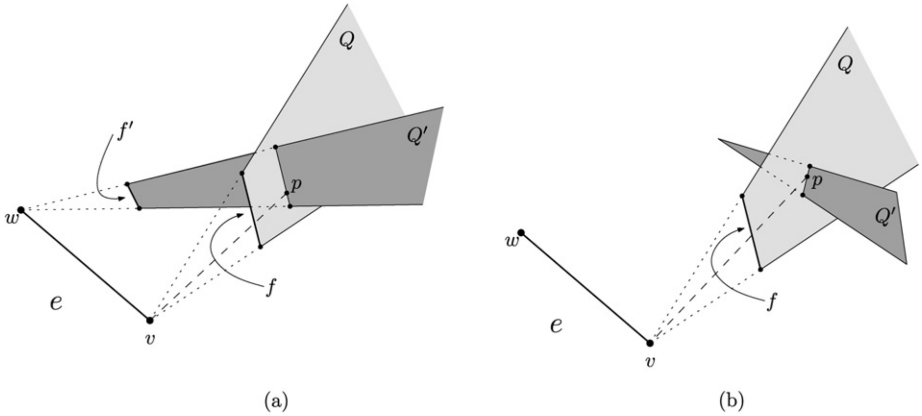


Fig. 4. Illustration of the proof of Lemma 9.

Definition 8. Given a 3D scene T and an edge $e = vw$ of T, we define $S(e, T)$ to be the following set of (convex) quadrilaterals:

$$S(e, T) = T \cup \mathcal{E} \cup \mathcal{V} \cup \mathcal{W},$$

where $\mathcal{E} = \{\mathcal{P}(e, u) \mid u \text{ vertex of } T\}$, $\mathcal{V} = \{\mathcal{P}(v, f) \mid f \text{ edge of } T\}$, and $\mathcal{W} = \{\mathcal{P}(w, f) \mid f \text{ edge of } T\}$.

If we split each quadrilateral in this set into a pair of triangles, $\mathrm{SVP}(e, T)$ is a single cell of the arrangement in 3D induced by $S(e, T)$ and consequently has complexity $\mathrm{O}(n^2 \log n)$ [19]. In the lemma below, we improve on this result by counting the number of features of this single cell on the quadrilaterals of $S(e, T)$ separately.

Lemma 9. *Let T be a 3D scene, and let e be an edge of T.* $\mathrm{SVP}(e, T)$ *has complexity* $\mathrm{O}(n^2\alpha(n))$.

Proof. The combinatorial complexity of $\mathrm{SVP}(e, T)$ is the complexity of the cell in question of the arrangement induced by $S(e, T)$. This complexity is the number of faces, which are part of the quadrilaterals of $S(e, T)$, plus the number of edges and vertices, which are intersections between, respectively, two or three of these quadrilaterals. We prove that this number is $\mathrm{O}(n^2\alpha(n))$ by showing that it is $\mathrm{O}(n\alpha(n))$ on each triangle in $\mathcal{E} \cup \mathcal{V} \cup \mathcal{W}$. Each feature on a (scene) triangle in T can be charged to a feature on one of the triangles of $\mathcal{E} \cup \mathcal{V} \cup \mathcal{W}$, since the triangles in T have disjoint interiors.

Let Q be a quadrilateral in $\mathcal{V}$; the cases for $\mathcal{W}$ and $\mathcal{E}$ are analogous. The remaining quadrilaterals from $S(e, T)$ induce a two-dimensional arrangement $\mathcal{A}_Q$ of line segments on Q. We count the number of features of $\mathcal{A}_Q$ that contribute to the complexity of $\mathrm{SVP}(e, T)$. Let f be the edge of T such that $Q = \mathcal{P}(v, f)$, and let p be a point in $\mathrm{SVP}(e, T) \cap Q$. The point p lies on a line segment that is the intersection of Q with another quadrangle Q' from $S(e, T)$. In Fig. 4, we show the cases where (a) Q' is from $\mathcal{W}$, and (b) Q' is a (scene) triangle from T. The third case is that Q' is from $\mathcal{E}$. In each case, let s be the line segment that connects the point p with v. Because p is in $\mathrm{SVP}(e, T)$, s intersects exactly one feature of T, namely the edge f.

No other quadrilaterals from $S(e, T)$ can intersect s, otherwise p would not see e completely. In other words, every point in $\mathcal{A}_Q$ for which the line segment that connects it with v intersects no other quadrilateral from $S(e, T)$ is in $\mathrm{SVP}(e, T) \cap Q$. If we consider the direction towards v as the upward direction, the boundary of $\mathrm{SVP}(e, T)$ on Q directly corresponds to the upper envelope of a set of $\mathrm{O}(n)$ Jordan curves that pairwise intersect at most once; see Fig. 5. The set of all points on Q that see e completely together form a face of $\mathrm{SVP}(e, T)$ that lies on Q (in fact, it is the only face of $\mathrm{SVP}(e, T)$ on Q), which is a connected region in $\mathcal{A}_Q$. Since the complexity of an upper envelope of such segments has complexity $\mathrm{O}(n\alpha(n))$ [23], the number of edges and vertices of $\mathrm{SVP}(e, T)$ that lie on Q is $\mathrm{O}(n\alpha(n))$. Hence, we have that $\mathrm{SVP}(e, T) \cap Q$ has complexity $\mathrm{O}(n\alpha(n))$ for each quadrilateral Q from $\mathcal{E} \cup \mathcal{V} \cup \mathcal{W}$, and the total complexity of $\mathrm{SVP}(e, T)$ is $\mathrm{O}(n^2\alpha(n))$. □

In conclusion, we have the following theorem.

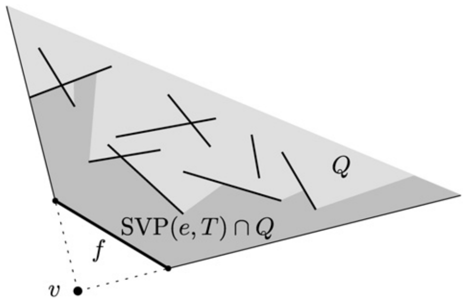


Fig. 5. Illustration of the proof of Lemma 9.

Theorem 10. *Let T be a 3D scene with n triangles, and let e be an edge of T. The strong visibility map* $\mathrm{SVM}(e, T)$ *has complexity* $\Omega(n^2)$ *and* $\mathrm{O}(n^2\alpha(n))$.

The proof of Lemma 9 gives us a way to efficiently compute the strong visibility map:

Algorithm COMPUTE_STRONG_VM
INPUT: A set T of n triangles in $\mathbb{R}^3$ and an edge e of T.
OUTPUT: $\mathrm{SVM}(e, T)$

1. Generate the set of quadrilaterals $S(e, T)$ as defined in Definition 8.
2. For each quadrilateral Q in $\mathcal{E} \cup \mathcal{V} \cup \mathcal{W}$:
   (a) Compute the intersections of Q with the other quadrilaterals in $S(e, T)$.
   (b) Compute the upper envelope of these line segments; this is the boundary of $\mathrm{SVP}(e, T)$ on Q.
   (c) Identify those features of the envelope that are induced by a triangle of T and store this information with that particular scene triangle.
3. For each triangle of T: compute the visible connected components from the information computed above. Output this set of connected components; this is $\mathrm{SVM}(e, T)$.

The first step trivially takes $\mathrm{O}(n)$ time, and step 2(a) $\mathrm{O}(n^2)$ time. In step 2(b), for each of the $\mathrm{O}(n)$ quadrilaterals from $\mathcal{E} \cup \mathcal{V} \cup \mathcal{W}$, we use the optimal algorithm of Hershberger [16] to compute the boundary of $\mathrm{SVP}(e, T)$ in $\mathrm{O}(n \log n)$ time, thus taking $\mathrm{O}(n^2 \log n)$ time overall. In step 2(c), we check the features that we computed in step 2(b) for intersection with triangles from T, which can trivially be done in $\mathrm{O}(n\alpha(n))$ time per quadrilateral in $S(e, T)$, so this step takes $\mathrm{O}(n^2\alpha(n))$ time in total. In step 3, we glue the connected components together; this can e.g. be done by a straightforward sweepline algorithm, which takes $\mathrm{O}(n^2\alpha(n) \log n)$.

Theorem 11. *Let T be a 3D scene with n triangles, and let e be an edge of T. The strong visibility map of e on T can be computed in* $\mathrm{O}(n^2\alpha(n) \log n)$ *time.*

4. The weak visibility map of an edge

4.1. Lower bounds

The weak visibility region of an edge in a polygon with holes can have complexity $\Omega(n^4)$, as was shown by Suri and O'Rourke [24]. In their construction, an edge light source and two rows of segments with suitably placed gaps A and B produce a quadratic number of narrow light cones that all mutually intersect. These intersections of cones result in $\Omega(n^4)$ illuminated vertices of the visibility region; see Fig. 6 for a schematic illustration of this construction.

We can create this same construction with a polyhedral terrain, where the 2D situation described above is replicated on the xy-plane, such that a single triangle contains all the $\Omega(n^4)$ intersections of the light cones. Thus, we have the following lemma.

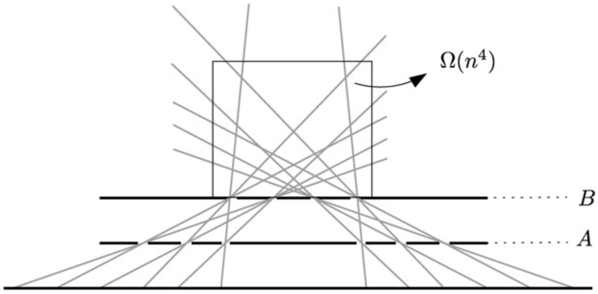


Fig. 6. A schematic illustration of the lower bound construction for the weak visibility region of an edge in a polygon with holes [24].

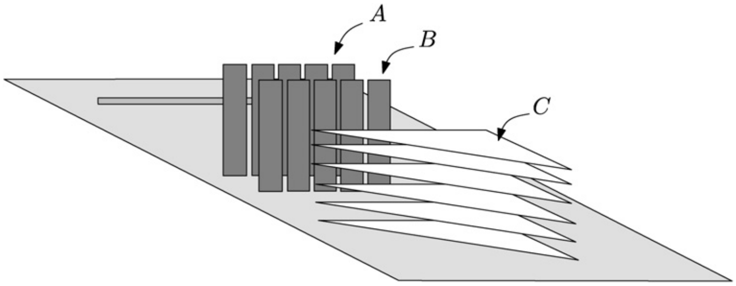


Fig. 7. A schematic display of a 3D scene in which the weak visibility map of an edge has complexity $\Omega(n^5)$.

Lemma 12. *The weak visibility map of an edge in a polyhedral terrain with n triangles can have* $\Omega(n^4)$ *vertices.*

For general scenes, we can improve this lower bound. We construct a scene such that every one of the $\Omega(n^2)$ light cones is unbounded in the z-direction, i.e., we have $\Omega(n^2)$ illuminated vertical planes, and each of the $\Omega(n^4)$ intersections of two cones is an illuminated vertical line. By placing a linear number of triangles vertically above each other, we can replicate the triangle that contains $\Omega(n^4)$ vertices of the visibility map $\Omega(n)$ times. Note that the triangles in this stack should be spaced in such a way that the light can extend through to the back of the stack. A schematic display of the lower bound construction for general 3D scenes is shown in Fig. 7. Due to the vertically stacked triangles in C, this construction cannot be built as a polyhedral terrain.

Lemma 13. *The weak visibility map of an edge in a 3D scene with n triangles can have* $\Omega(n^5)$ *vertices.*

4.2. Upper bounds

Throughout this section, we are given a 3D scene T with n triangles and a designated edge e of T that acts as a light source. We prove upper bounds on the worst-case complexity of the shadow that e induces, together with the scene triangles, onto different features of T. In Section 4.2.1 we prove a bound of $\Theta(n^2)$ for the complexity of the shadow on an edge of T, and in Section 4.2.2 we prove $\Theta(n^4)$ for the complexity of the shadow on a triangle of T. Note that this last result, together with Lemma 13, implies that the visibility map of an edge in a general 3D scene has complexity $\Theta(n^5)$.

First, we look at the shadow that an edge light source e induces on a single point p on a 3D scene, which is equivalent to asking whether p is illuminated or not. Let Δ be the triangle that has p as its apex and the edge e as its base. We compute the set S of line segments that are intersections of the $O(n)$ triangles of T with Δ. This can be done in $O(n)$ time. Then, we project S onto e with respect to p, and compute whether these projections, which are intervals on e, cover e completely. We can do this by using an $O(n \log n)$ sweepline algorithm. It is easy to see that p is not illuminated by e if and only if e is completely covered by the projections of S, which can be checked while we sweep the intervals.

The above bound on deciding whether or not a point is illuminated is in fact tight. The $\Omega(n \log n)$ lower bound follows from an adaptation of the lower bound proof of Klee's Measure Problem: given a set of n (possibly overlapping)

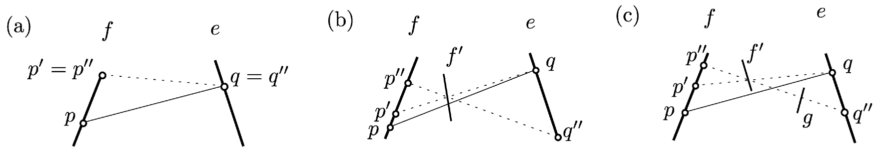


Fig. 8. Three cases that arise in the proof of Lemma 14.

intervals $[a_i, b_i]$, compute the measure of their union. This problem is known to have a lower bound of $\Omega(n \log n)$ in the algebraic decision tree model [10], and the proof can easily be adjusted to provide an $\Omega(n \log n)$ lower bound for this illumination problem.

4.2.1. Complexity on an edge

By the lower bound construction in [24], the shadow map restricted to an edge can have complexity $\Omega(n^2)$, since there can be a quadratic number of light cones that intersect the edge in distinct places. The aspect graph gives us a trivial upper bound on the shadow of $O(n^3)$, because that is the maximum number of different views of the light source e seen from f. We show that the $\Omega(n^2)$ lower bound is tight.

Lemma 14. *Let T be a 3D scene with n triangles, and let e and f be two edges of T. If e acts as a light source, then the shadow that T casts on f has complexity* $O(n^2)$.

Proof. If no point on f is illuminated, then the shadow map on f has complexity $O(1)$ and the lemma is trivially true. Otherwise, let p be an arbitrary illuminated point on f, and let q be a point on e that illuminates p; the open line segment pq does not intersect T. Now, we start moving p over f, in either of the two directions. We can keep moving p either until we reach the end of f, see Fig. 8(a), or until the line segment pq intersects an edge f' of T. By construction, the point p' we determine in this way is illuminated, namely by q. Next, we move the point p' over f in the same direction as we moved p, while maintaining contact with f'. This corresponds to sweeping the plane defined by p' and f' over e. We keep moving until we reach the end of f, which is similar to the case in Fig. 8(a), we reach the end of either e or f' (see Fig. 8(b)), or touch a second edge g of T (see Fig. 8(c)). In each case, we determine a point p'' on f that is illuminated by a point q'' on f.

Observe that p'' is (a) an endpoint of f, (b) an extremal point on f of one of the connected components of the regulus $\mathcal{R}(e, f', f)$ for some edge f' of T, or (c) a point of intersection of f with the regulus $\mathcal{R}(e, f', g)$ or the regulus $\mathcal{R}(e, g, f')$, where f' and g are edges of T. Analogously, we can find a similar illuminated point if we start by moving p in the opposite direction. In the remainder of the proof, we bound the number of such points on f.

For every pair of edges (f', g) of T, we consider the regulus $\mathcal{R}(e, f', g)$. If this set exists, then it intersects the edge f at most in a constant number of points (see Section 2). Consider the arrangement $\mathcal{A}(f)$ on f that is induced by the intersections of the reguli $\mathcal{R}(e, f', g)$ with e, for all ordered pairs (f', g) of edges of T, with $f' \neq g$. Since there are $O(n^2)$ such reguli, the arrangement $\mathcal{A}(f)$ has complexity $O(n^2)$. By the discussion above, every interval $i \in \mathcal{A}(f)$ is completely illuminated or completely dark. Thus, $O(n^2)$ is an upper bound on the complexity of the shadow map of e on f. □

4.2.2. Complexity on a triangle

By the lower bound construction in [24], the shadow on a triangle t of a 3D scene can have complexity $\Omega(n^4)$. The aspect graph gives us a trivial upper bound on the shadow of $O(n^6)$. Below, we prove that the lower bound is tight.

Lemma 15. *Let T be a 3D scene with n triangles, let e be an edge of T, and let t be a triangle of T. If e acts as a light source, then the shadow that T casts on t has complexity* $O(n^4)$.

Proof. If no point on t is illuminated, then the shadow has complexity $O(1)$ and the lemma is trivially true.

Otherwise, let p be an arbitrary illuminated point on t, and let q be a point on e that illuminates p; the open line segment pq does not intersect T. Now we choose any arbitrary straight line segment s through p that divides t in two. By the argumentation in the proof of Lemma 14, we can find an illuminated interval on s, not necessarily maximal, that contains p and is bounded on both sides by either an endpoint of s, or by a point that is the intersection of s

with a line that goes through e as well as two edges f and g of T. Because s was chosen arbitrarily, we can find a two-dimensional illuminated region on t that contains p and is bounded by curves that are intersections of T with one or more reguli $R(e, f_i, g_j)$, where f_i and g_j are edges of T.

Consider the arrangement $\mathcal{A}(t)$ on t that is induced by the curves that are intersections of all possible reguli $\mathcal{R}(e, f, g)$ with t, for all ordered pairs (f, g) of edges of T, with $f \neq g$. Because there are $O(\binom{n}{2})$ edge pairs (f, g), there are $O(n^2)$ such curves, and this arrangement on t has complexity $O(n^4)$. By the discussion above, every cell $c \in \mathcal{A}(t)$ that contains at least one illuminated point, must be completely illuminated. Every cell $c' \in \mathcal{A}(e)$ that contains at least one shadow point, by the same arguments, must lie completely in the dark. Thus, $O(n^4)$ is an upper bound on the complexity of the shadow on t. □

Putting Lemmas 12, 13 and 15 together yields the following theorem.

Theorem 16. *Let T be a 3D scene with n triangles, and let e be an edge T. Then* $\mathrm{WVM}(e, T)$ *has complexity* $\Theta(n^5)$ *in the worst case. If T is a polyhedral terrain, then the complexity of* $\mathrm{WVM}(e, T)$ *is* $\Omega(n^4)$ *and* $O(n^5)$.

4.3. Algorithm

To compute the shadow of T that is cast onto an edge f by the illuminating edge e, we use a sweep algorithm. Our algorithm is very similar to the one described in [6]; however, we sweep the *illuminated* edge instead of the *illuminating* one. The reason for this adjustment is that, in this way, we immediately obtain the subdivision of f into illuminated and dark points, i.e., the shadow map of e on f.

First, we compute the set R of all reguli $R(e, f', g)$ for all ordered edge pairs (f', g) of T; there are $O(n^2)$ such reguli. Then we compute $\mathcal{A}(R, f)$, the arrangement on f induced by the intersections of the reguli in R with f. For one of the endpoints of f, say p_0, we compute the intersections of the triangles of T with the triangle Δ_{ep_0}. We project these intersections onto e; we now have a set S of $O(n)$ intervals on e, in these intervals either every point illuminates p_0 or none of them does. Now we move the point p_t, which initially is the point p_0, over the edge f, which corresponds to sweeping the triangle Δ_{ep_t} through T. In the meantime, we maintain the set S of (projected) intervals on e. The *event points* of the sweep are the points where (i) a new interval (either illuminating or not) appears, (ii) an existing interval disappears, or (iii) two endpoints of different intervals change order. Because the intervals correspond directly to (sets of) triangles of T, it is easy to see that the event points are exactly the interval boundaries of $\mathcal{A}(R, f)$. An interval i of $\mathcal{A}(R, f)$ is illuminated if there is a point on e that illuminates any point $p \in i$, and is in the shadow otherwise.

Computing $\mathcal{A}(R, f)$ takes $O(n^2 \log n)$ time, as we compute and sort the intersections of a quadratic number of reguli with an edge, and every regulus intersects an edge a constant number of times. Computing the set S of intervals on e that illuminate p_0 takes $O(n \log n)$ time, as we showed at the beginning of Section 4.2. The number of event points we encounter during the sweep is $O(n^2)$ [6] and at every event point we need to update the status structure in a constant number of places, since there are exactly two edges of T associated with each event point. With an appropriate data structure, e.g., an augmented binary search tree, this step, and thus the complete algorithm, takes $O(n^2 \log n)$ time.

Lemma 17. *Let T be a 3D scene with n triangles, and let e and f be two edges of T. We can compute the part of* $\mathrm{WVM}(e, T)$ *restricted to f in* $O(n^2 \log n)$ *time.*

The shadow of T cast onto a triangle t of T by the illuminating edge e can be computed by a sweep algorithm as well. First, we need an extra lemma. Since two algebraic curves of constant degree in the plane have at most a constant number of intersections, it is easy to verify the lemma below (the proof is basically similar to the proof of Lemma 14 and the analysis in [6]).

Lemma 18. *Let e be an edge of a 3D scene T with n triangles, and let c be an algebraic curve of constant degree on a triangle t of T. If e acts as a light source, then the shadow that T casts on c has complexity* $O(n^2)$ *and can be computed in* $O(n^2 \log n)$ *time.*

The first step in computing the shadow map of an edge e on a triangle t is to generate the set $S(t)$ of curves of intersection of the reguli $\mathcal{R}(e, f, g)$ with t, for all ordered edge pairs (f, g) of T. Next, we compute the subdivision of t induced by $S(t)$, denoted by $\mathcal{A}(S(t))$. Let K be the complexity of this arrangement, which is $O(n^4)$ in the worst case. The arrangement $\mathcal{A}(S(t))$ can be computed in $O(n^2 \log n + K \log n)$ time [13]. We execute the $O(n^2 \log n)$ algorithm that computes the shadow map on an algebraic curve *twice* for every curve c that appears in $\mathcal{A}(S(t))$; we compute curves c' and c'' that, symbolically, lie infinitesimally close to c on either side. By computing the shadow maps on c' and c'' we can determine whether the adjacent cells of $\mathcal{A}(S(t))$ are *illuminated* or *dark*. If, during the construction of $\mathcal{A}(S(t))$, we store with every edge from which curve it originates, this cell labeling can be performed in $O(K \log n)$ time. Although the output-sensitive construction of $\mathcal{A}(S(t))$ will usually mean a more efficient algorithm in practice, it does not imply that we can compute the visibility map in an output-sensitive way, since not every regulus necessarily separates visible and invisible points. Thus, the running time of the above algorithm is $O(n^4 \log n)$ in the worst case.

Lemma 19. *Let T be a 3D scene with n triangles, let e be an edge of T, and let t be a triangle of T. We can compute the part of the weak visibility map of e on t in $O(n^4 \log n)$ time.*

Now it is trivial to extend the above procedure to compute the complete weak visibility map of e on T.

Algorithm COMPUTE_WEAK_VM
INPUT: An edge e and a 3D scene T.
OUTPUT: $\mathrm{WVM}(e, T)$

1. For every triangle t of T, compute the shadow induced by e on t in the way described above.
2. Return the set of all illuminated connected components, which is $\mathrm{WVM}(e, T)$.

We now have obtained the following result.

Theorem 20. *Let T be a 3D scene with n triangles, and let e be an edge of T. The weak visibility map $\mathrm{WVM}(e, T)$ can be computed in $O(n^5 \log n)$ time.*

5. The triangle visibility map

We also consider the problem where the light source is neither a point nor a line segment, but a triangle in 3D. Using the techniques and ideas of the previous two sections, we show bounds on the worst-case combinatorial complexity of the two types of visibility maps for a triangle in 3D.

5.1. The strong visibility map of a triangle

The strong visibility map of a triangle t in a 3D scene T is the subdivision of the triangles of T into points that completely see t and points that do not. Like in Section 3, we bound the combinatorial complexity of the strong visibility polyhedron $\mathrm{SVP}(t, T)$, which also gives us an upper bound on the complexity of the strong visibility map $\mathrm{SVM}(t, T)$.

Let e_1, e_2, and e_3 be the three edges of t. We define $S(t, T)$ to be the set of quadrilaterals $\bigcup_{i=1..3} S(e_i, T)$, with $S(e_i, T)$ as in Definition 8. By the results in [19] and a similar argumentation as in the proof of Lemma 9, $\mathrm{SVP}(t, T)$ is a single cell in the three-dimensional arrangement induced by $S(t, T)$ with complexity $O(n^2\alpha(n))$. The algorithm to compute $\mathrm{SVM}(t, T)$ is similar to the one described in Section 3, which gives us the result below.

Theorem 21. *Let T be a 3D scene with n triangles, and let t be a triangle of T. Then $\mathrm{SVP}(t, T)$ and $\mathrm{SVM}(t, T)$ have complexity $\Omega(n^2)$ and $O(n^2\alpha(n))$ and can be computed in $O(n^2\alpha(n) \log n)$ time.*

5.2. The weak visibility map of a triangle

As we did in Section 4, we distinguish the two cases where the scene T is a terrain, and where T is a general 3D scene.

5.2.1. Polyhedral terrains

The lower bound construction where the visibility map of an edge has $\Omega(n^4)$ vertices (see Fig. 6) trivially yields the same lower bound for a triangle light source.

The $O(n^5)$ upper bound on the complexity of the weak visibility map of an edge is also an upper bound for the weak visibility map of a triangle t. To verify this, suppose there is a point p in the weak visibility map of $\mathrm{WVM}(t)$, and let q be a point on t that sees p. Then we can rotate the line-of-sight pq vertically upwards around p, until we obtain a line-of-sight pq' that is free from intersections with T, such that the point q' lies on one of the edges of t. Hence, $\mathrm{WVM}(t) \subseteq \bigcup_{i=1..3} \mathrm{WVM}(e_i)$. Trivially, it also holds that $\mathrm{WVM}(t) \supseteq \bigcup_{i=1..3} \mathrm{WVM}(e_i)$. Because on every triangle of T, the complexity of $\bigcup_{i=1..3} \mathrm{WVM}(e_i)$ is bounded by the complexity of the arrangement induced by three sets of $O(n^2)$ reguli, $\mathrm{WVM}(t, T)$ is $\Theta(n^4)$ on a single triangle and $O(n^5)$ overall.

Theorem 22. *Let T be a polyhedral terrain, and let t be a triangle of T. Then* $\mathrm{WVM}(t, T)$ *has complexity* $\Omega(n^4)$ *and* $O(n^5)$.

5.2.2. General 3D scenes

Now, we consider the complexity of the weak visibility map of a triangle t on a general 3D scene T. Recall from Definition 4 in Section 2 that the aspect graph is the subdivision of space into connected regions where the view is combinatorially the same. In other words, all points in a single cell of the aspect see exactly the same features of T. Obviously, in every cell the triangle t is either visible or invisible. Thus, the complexity of the aspect graph *restricted to the triangles* of T is an upper bound on $\mathrm{WVM}(t, T)$. The complete aspect graph, i.e., the subdivision of $\mathbb{R}^3$, is defined by $O(n^3)$ critical surfaces [7]. For every triangle, this gives us a set of $O(n^3)$ critical curves, which in turn induces an arrangement of complexity $O(n^6)$ on a single triangle, and thus $O(n^7)$ on T. Hence, we have the following lemma.

Lemma 23. *Let T be a 3D scene and let t be a triangle of T. Then* $\mathrm{WVM}(t, T)$ *has complexity* $O(n^7)$.

In the remainder of this section, we show that $O(n^7)$ is tight, by extending the $\Omega(n^5)$ lower bound construction of Fig. 7. The basic idea of the extension is the following: first, we 'replace' A by a row giving $\Omega(n^2)$ gaps, which produces $\Omega(n^3)$ illuminated vertical planes, and thus we get $\Omega(n^6)$ illuminating vertical lines that each intersect $\Omega(n)$ triangles. In this way, we create $\Omega(n^7)$ vertices. We describe this construction in detail.

In order to construct a row with $\Omega(n^2)$ gaps that let the light that is emitted by t pass through without using more than $O(n)$ triangles, we first make the following observation. Each of the illuminated vertical planes of the lower bound of Fig. 7 is uniquely determined by two gaps: one in A and one in B. Besides the obvious restriction that the light source should be located on the opposite side of A than B, these illuminated planes do not depend on the location of the light source. Because we now have an extra degree of freedom, we have the opportunity to make $\Omega(n^2)$ gaps by building a grid G instead of the row A. This idea is illustrated in Fig. 9(a). The important fact here is that, if the holes in the grid degenerate to points, every hole has a unique x-coordinate.

If we allow enough distance between G and B, the holes in G basically act as point light sources, emitting cones of light. Now, just like in the old construction, each of the holes in G interacts with each of the gaps in B, producing $\Omega(n^3)$ illuminated vertical planes in total; see Fig. 9(b).

The total construction is illustrated schematically in Fig. 10. First, we have the illuminating triangle t, followed by the grid G, then the row of vertical slabs B, and finally the set C of vertically stacked triangles. To ensure that we indeed get $\Omega(n^6)$ illuminated vertical lines, we place a copy of G and B on the xy-plane such that they generate $\Omega(n^3)$ illuminated vertical planes from a different direction; the light source must be large enough to provide light for both constructions. This gives us another set of $\Omega(n^3)$ illuminated vertical planes, intersecting the $\Omega(n^3)$ planes that are produced by the first copy.

Note that we have to make sure that all the vertices we create on the white triangles of Fig. 10 are indeed vertices of the weak visibility map of t. To achieve this, we add a constant number of very large triangles (not shown in Fig. 10) to block all light coming from the light source that does not pass through the holes in G and B or the corresponding holes in the orthogonal copy of the construction. Concluding, we have the following theorem.

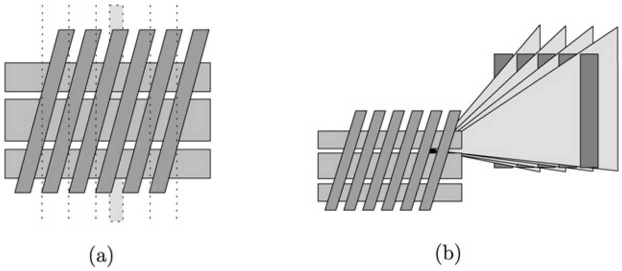


Fig. 9. (a) A vertical grid with $\Omega(n^2)$ holes that allow light to pass through. (b) Every combination of a hole in the grid and a gap between two vertical slabs induces $\Omega(n)$ illuminated planes.

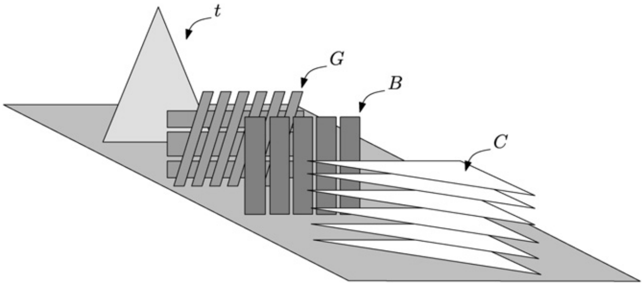


Fig. 10. The total lower bound construction. The copy of G and B that induces another $\Omega(n^3)$ illuminated planes is not shown.

Theorem 24. *Let T be a 3D scene with n triangles, and let t be a triangle of T. Then* $\mathrm{WVM}(t, T)$ *has complexity* $\Theta(n^7)$ *in the worst case.*

5.3. Algorithm

We present an $O(n^7 \log n)$ time algorithm to compute $\mathrm{WVM}(t, T)$. Take one triangle Δ from T and select and all triangles of T that intersect the convex hull of t and Δ. This gives a set T' of triangles that lie (entirely or partially) in between t and Δ. Next, we compute the arrangement $\mathcal{A}$ of all reguli $\mathcal{R}(e, f, g)$ on Δ, where e, f, g are edges from triangles in $T' \cup t$. This two-dimensional arrangement $\mathcal{A}$ is induced by $O(n^3)$ curves on Δ. As in Section 4.3, we can compute $\mathcal{A}$ in an output-sensitive way, i.e., in $O((n^3 + K) \log n)$ time, where K is the complexity of $\mathcal{A}$, which is $O(n^6)$ in the worst case. However, again this will not give us an output-sensitive algorithm to compute the visibility map, since not all reguli that occur in $\mathcal{A}$ actually contribute to $\mathrm{WVM}(t, T)$.

Choose a point p in some cell c in $\mathcal{A}$, and compute the arrangement $\mathcal{E}$ of edges of T' projected perspectively from p onto the plane supporting t. This is also the arrangement of projected triangles of T' if they were transparent. The two-dimensional arrangement $\mathcal{E}$ is induced by $O(n)$ line segments and can be computed in $O(n^2)$ time [13]. Because $\mathcal{A}$ corresponds directly to the aspect graph on Δ, the combinatorial structure of $\mathcal{E}$ is the same for any point in c. It only changes when we move p in $\mathcal{A}$ such that p crosses a regulus $\mathcal{R}(e, f, g)$ and enters a cell c' that is adjacent to c. In $\mathcal{E}$, the following happens. The projected edges of e, f, g must have had subedges that formed a triangle in $\mathcal{E}$, which collapses when p reaches $\mathcal{R}(e, f, g)$. When p enters c', an inverted triangle appears between subedges of the projections of e, f, g. Alternatively, two pairs of faces of $\mathcal{E}$ can split or merge when p crosses a regulus that is defined by two edges of the same triangle.

After computing $\mathcal{E}$ for point p in c, we compute some more information. For every cell in $\mathcal{E}$, we compute how many triangles cover that cell, and store this information with the cell. We also maintain whether the cell lies inside or outside t; recall that $\mathcal{E}$ lies in the plane supporting t. For every projected edge e, we maintain a balanced binary search tree $\mathcal{T}_e$ on its intersections with other projected edges. Leaves storing these intersections also contain a pointer into the structure for $\mathcal{E}$; it provides efficient access. Finally, we maintain a global count I of the number of cells of $\mathcal{E}$ that lie inside t and are covered by zero projected triangles, i.e., the number of cells of $\mathcal{E}$ that illuminate the cell c. If for a cell c of $\mathcal{A}$ the global count is positive, then it is illuminated; if the global count is zero it is dark. See Fig. 11 for an illustration of the various structures that the algorithm incorporates.

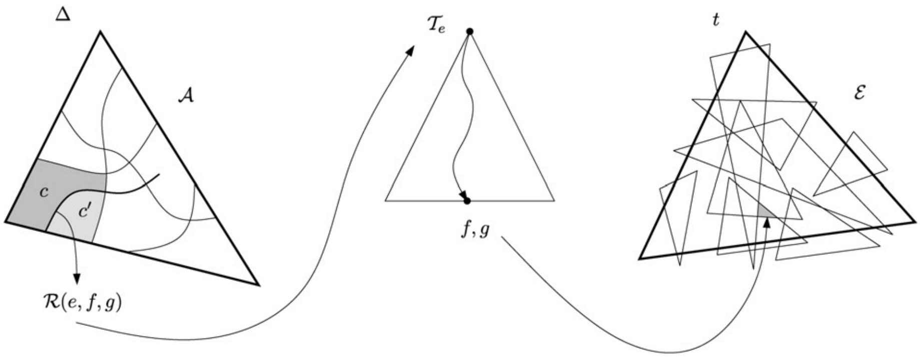


Fig. 11. Illustration of the algorithm that computes the part of WVM(t, T) on a triangle Δ.

All of the above information can be updated when we go from cell c to a neighboring cell c' in $O(\log n)$ time. We must exchange a pair of leaves in three binary search trees, we have to make local changes to $\mathcal{E}$ for the collapsing and/or appearing faces, we must determine the number of projected triangles that cover the new cells in $\mathcal{E}$, and we have to update the global count I of 'illuminating' cells of $\mathcal{E}$. The number of projected triangles that cover the new faces in $\mathcal{E}$ has the corresponding value of the disappearing face, plus or minus 1 or 3. This can be determined from the edges e, f, g of T'. The value of I can decrease or increase by 1, or stay the same. Only the changes of the binary trees takes $O(\log n)$ time, the other updates require only $O(1)$ time, after which we can label c' either illuminated or dark. If $\mathcal{A}$ has complexity K, which is $O(n^6)$ in the worst case, the total update time if we traverse all cells of $\mathcal{A}$ is $O(K \log n)$, after which we have obtained the weak visibility map of t on Δ. Repeating this procedure for all n triangles of T gives us WVM(t, T).

Theorem 25. *Let T be a 3D scene with n triangles, and let t be a triangle of T. The weak visibility map* WVM(t, T) *can be computed in* $O(n^7 \log n)$ *time.*

6. Concluding remarks

We presented lower and upper bounds on the combinatorial complexities of various types of visibility maps of edges and triangles in a 3D scene. As can be seen in Table 1, there are a few gaps left between lower and upper bounds. Finally, considering the discussion in Sections 4.3 and 5.3 about the output-sensitive computation of arrangements of reguli, it would be interesting to find algorithms that are indeed output-sensitive with respect to the complexity of the visibility map in question.

Acknowledgements

The authors thank an anonymous referee for valuable comments, especially those that improved the results in Section 3.

References

[1] P.K. Agarwal, M. Sharir, On the number of views of polyhedral terrains, Discrete Comput. Geom. 12 (1994) 177–182.
[2] T. Asano, S.K. Ghosh, T.C. Shermer, Visibility in the plane, in: Handbook of Computational Geometry, North-Holland, Amsterdam, 2000 (Chapter 19).
[3] M. Bern, D. Dobkin, D. Eppstein, R. Grossman, Visibility with a moving point of view, in: Proc. 1st ACM-SIAM Sympos. Discrete Algorithms, 1990, pp. 107–117.
[4] H. Brönnimann, H. Everett, S. Lazard, F. Sottile, S. Whitesides, Transversals to line segments in three-dimensional space, Discrete Comput. Geom. 34 (3) (2005) 381–390.
[5] N. Chin, S. Feiner, Fast object-precision shadow generation for areal light sources using BSP trees, in: Proc. Sympos. Interactive 3D Graphics, 1992, pp. 21–30.
[6] N. Coll, M. Fort, J. Sellarès, Approximate multi-visibility map computation, in: Abstr. Europ. Workshop Comput. Geom., 2005, pp. 97–100.
[7] M. de Berg, D. Halperin, M. Overmars, M. van Kreveld, Sparse arrangements and the number of views of polyhedral scenes, Internat. J. Comput. Geom. Appl. 7 (1997) 175–195.

[8] F. Duguet, G. Drettakis, Robust epsilon visibility, in: Proc. ACM SIGGRAPH'02, 2002, pp. 567–575.
[9] H. Edelsbrunner, E.P. Mücke, Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms, ACM Trans. Graph. 9 (1) (1990) 66–104.
[10] M.L. Fredman, B. Weide, On the complexity of computing the measure of $\bigcup[a_i, b_i]$, Commun. ACM 21 (1978) 540–544.
[11] S. Ghali, E. Fiume, H.-P. Seidel, Object-space, connectivity-preserving, shadow computation, in: Proc. Conf. Vis. Mod. Visualiz. (VMV-00), 2000, pp. 281–289.
[12] Z. Gigus, J. Canny, R. Seidel, Efficiently computing and representing aspect graphs of polyhedral objects, IEEE Trans. Pattern Anal. Mach. Intell. 13 (6) (1991) 542–551.
[13] D. Halperin, Arrangements, in: J.E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press LLC, Boca Raton, FL, 2004, pp. 529–562 (Chapter 24).
[14] J.-M. Hasenfratz, M. Lapierre, N. Holzschuch, F. Sillion, A survey of real-time soft shadows algorithms, Comp. Graphics Forum 22 (4) (2003) 753–774.
[15] J. Heintz, T. Reico, M.-F. Roy, Algorithms in real algebraic geometry and applications to computational geometry, in: J.E. Goodman, R. Pollack, W. Steiger, (Eds.), Discrete Comput. Geom.: Papers from the DIMACS Special Year, 1991, pp. 137–163.
[16] J. Hershberger, Finding the upper envelope of n line segments in $O(n \log n)$ time, Inform. Process. Lett. 33 (1989) 169–174.
[17] J.J. Koenderink, A.J. van Doorn, The internal representation of solid shape with respect to vision, Biological Cybernetics 32 (1979) 211–216.
[18] M. McKenna, Worst-case optimal hidden-surface removal, ACM Trans. Graph. 6 (1987) 19–28.
[19] E. Moet, C. Knauer, M. van Kreveld, Visibility maps of segments and triangles in 3D, Technical Report UU-CS-2005-049, Universiteit Utrecht.
[20] J. O'Rourke, Visibility, in: J.E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press, Boca Raton, FL, USA, 2004 (Chapter 28).
[21] H. Plantinga, C.R. Dyer, Visibility, occlusion, and the aspect graph, Internat. J. Comput. Vision 5 (2) (1990) 137–160.
[22] H. Pottmann, J. Wallner, Computational Line Geometry, Springer-Verlag, Berlin, 2001.
[23] M. Sharir, P.K. Agarwal, Davenport–Schinzel Sequences and Their Geometric Applications, Cambridge University Press, New York, 1995.
[24] S. Suri, J. O'Rourke, Worst-case optimal algorithms for constructing visibility polygons with holes, in: Proc. 2nd Annu. ACM Sympos. Comput. Geom., 1986, pp. 14–23.
[25] S.J. Teller, Computing the antipenumbra of an area light source, Comput. Graph. 26 (4) (1992) 139–148.
[26] C. Wang, B. Zhu, Three-dimensional weak visibility: Complexity and applications, Theor. Comput. Sci. 234 (2000) 219–232.
[27] A. Woo, P. Poulin, A. Fournier, A survey of shadow algorithms, IEEE Comput. Graph. Appl. 10 (6) (1990) 13–32.